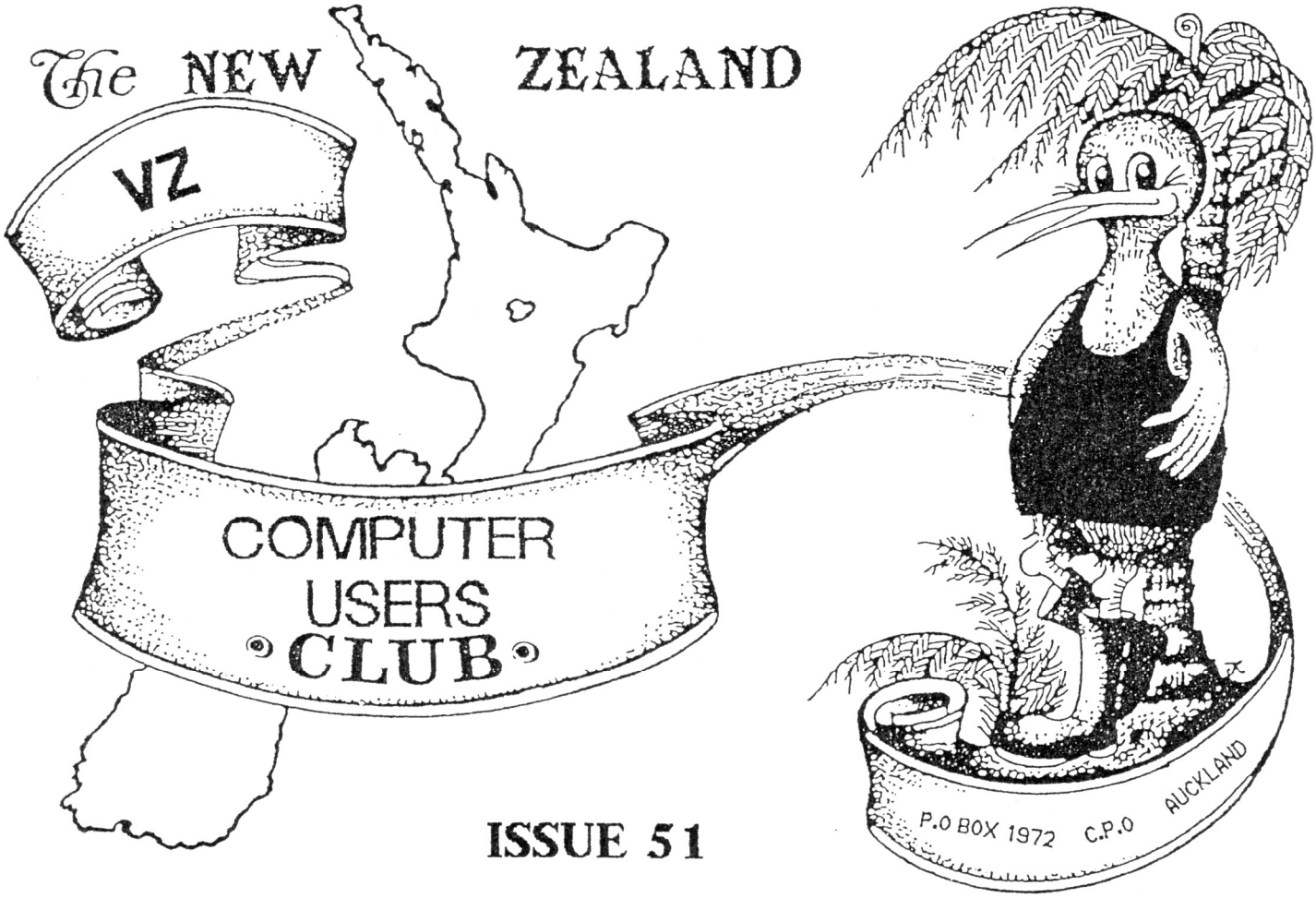
VZ LINK magazine



ISSUE 51

JANUARY 1991

FROM ME TO YOU

Well, here we are again with another jammed packed issue of your favourite newsletter. I hope you all had a very fruitful Christmas and are enjoying a good break before getting back to the grind.
Thank you, to all those people who sent me Christmas cards, I hope I didn't forget to send anybody one.

This issue, being the first of a new year, we are using an all new cover.
I hope you like it as much as I do. You may see this cover a lot more, because I will be using it when I don't have any other art work available.

I have packed a little more into this issue so as to tie up some loose ends.
It will not be the norm, unless you write some more items for the newsletter of course.

This year I hope to keep up the high standards of this newsletter.
To do this will involve as many of you as possible, so start thinking about what you can send in for the newsletter. It doesn't have to be only computer related you know.

Also this year I hope to start having Core group meetings at least once a fortnight. These will be on a small scale and you just come if you can.
More details in a future issue about this.

THIS ISSUE

We have a program which I hope you may all find useful at some stage.
Base Converter by Dave Mitchell will allow you to convert Decimal to Hex etc.
Ross Woods lists VZ mods which have appeared in a couple of magazines.
We continue the MAZE of DOOM program. A long way to go yet, so get typing or you will fall behind and get lost in the maze.
Robert Quinn asks " is this a video disk? ".
Don't ask me, I don't give the items their titles.
We also have a software review and some more programming hints for you.
Bob Kitch explains the USR function, I hope this makes it a little clearer for you.
Potpourri tells us all how to clean our contacts. This needs doing I can tell you.
Lastly we revisit the mailing list to clear up a problem and we have a addendum of graphic notes.

PROGRAM PROBLEM

The top secret program in the November issue was so secret that it would not work. As soon as I contact Steve Dale I will find out what is wrong.
I will then let you all know in due course.

P.O BOX 1972 C.P.O AUCKLAND N.Z

(C) AUCKLAND VZ 300/200 COMPUTER USERS CLUB (1986)
NEW ZEALAND VZ 300/200 COMPUTER USERS CLUB (1989)

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT.
CONTRIBUTED PROGRAMS & ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT IT IS FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.
COPYRIGHT IS ALWAY'S RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR ANY BREECHING OF COPYRIGHT CAUSED BY THE PUBLICATION OR USE OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME IF YOU HAVE PROBLEMS.
PHONE 761-451 AFTER 6.00 PM.

BASE CONVERTER BY DAVE MITCHELL

```
0 '*****************************************************
2 '*  CONVERT DEC TO HEX TO BINARY TO LO-BYTE AND HI-BYTE  *
4 '*  MODIFIED AND ENHANCED FOR THE VZ  BY  DAVE MITCHELL  *
6 '*****************************************************
8 :
10 POKE30862,0:POKE30863,114:POKE29184,243:POKE29185,201
20 POKE30777,1:Z$="                                    "
30 CLS:PRINT" THIS UTILITY WILL DISPLAY THE   EQUIVALENT";
40 PRINT" VALUES OF DECIMAL,   HEX AND BINARY ON THE SCREEN"
50 PRINT" AND TO THE PRINTER."
60 PRINT@164,"[illegible] ENTER DECIMAL VALUE"
70 PRINT@228,"[illegible] ENTER BINARY  VALUE"
80 PRINT@292,"[illegible] ENTER HEX     VALUE"
90 PRINT@356,"[illegible] ENTER L.S.B.& M.S.B."
100 PRINT@420,"[illegible]"
110 AA$=INKEY$:A$=INKEY$:IFA$=""THEN110
120 SOUND30,1
130 IFA$="A"THENGOSUB180:GOSUB230:GOTO110
140 IFA$="B"THENGOSUB180:GOSUB240:GOTO110
150 IFA$="C"THENGOSUB180:GOSUB250:GOTO110
160 IFA$="D"THENGOSUB180:GOSUB720
170 GOTO110
180 PRINT@0,Z$:PRINT@32,Z$:PRINT@64,Z$:PRINT@96,Z$
190 PRINT@0,"DECIMAL =":PRINT@32,"HEX     ="
200 PRINT@64,"BINARY  =":PRINT@96,"L.S.B   ="
210 PRINT@113,"M.S.B. =":RETURN
220 PRINT@448,Z$:RETURN
230 GOSUB410:GOSUB540:GOSUB260:RETURN
240 GOSUB470:GOSUB540:GOSUB260:RETURN
250 GOSUB620:GOSUB420:GOSUB260:RETURN
260 PRINT@480,"PRINT OUT [illegible]";
270 AA$=INKEY$:A$=INKEY$:IFA$<>"Y"ANDA$<>"N"THEN270
280 SOUND30,1
290 PRINT@480,"                    ";
300 IFA$="N"THEN400
310 IFC<=32767THEN330
320 LPRINT"DECIMAL VALUE ="C;"    ( ";INT(C-65536);")":GOTO340
330 LPRINT"DECIMAL VALUE ="C
340 LPRINT" HEX    VALUE = ";H1$
350 LPRINT" BINARY VALUE = ";C$
360 LPRINT"LEAST SIGNIFICANT BIT=";E2
370 LPRINT"MOST SIGNIFICANT BIT =";E1
380 REM LINE 309 LPRINTSTRING$(20,46)
390 LPRINTSTRING$(20,46)
400 C$="":H1$="":RETURN
410 B$="":PRINT@448,"ENTER DECIMAL VALUE";:INPUTD:C=D:GOSUB220
420 X=USR(0):B=D/2:D=INT(B):IFB=DTHENB$="0"+B$ELSEB$="1"+B$
430 IFD>0THEN420ELSEPRINT@74,B$
440 C$=B$:B$=""
450 PRINT@9,C:IFC<=32767THENRETURN
460 PRINT@20,CHR$(104);" ";C-65536;CHR$(105):RETURN
470 D=0:PRINT@448,"ENTER BINARY ";:INPUTB$:C$=B$:GOSUB220
480 X=USR(0)
490 FORI=LEN(B$)TO1STEP-1:IFMID$(B$,I,1)="1"THEND=D+2^(LEN(B$)-I)
500 NEXT:PRINT@74,C$
510 IFD<=32767THEN530
520 PRINT@20,CHR$(104);" ";INT(D-65536);CHR$(105)
530 PRINT@9,D:C=D:D=0:B$="":I=0:RETURN
540 X=USR(0):AD=C
550 B=INT(AD/256):E1=INT(B):GOSUB580:H$=B$
560 B=AD-(256*B):E2=INT(B):GOSUB580:H$=H$+B$
570 PRINT@42,H$:H1$=H$:PRINT@105,E2:PRINT@121,E1:RETURN
```

CONTINUED ON NEXT PAGE

BASE CONVERTER CONT. - - - - - -

```
580 H=INT(B/16):L=B-(H*16):B$=""
590 D=H:GOSUB600:D=L:GOSUB600:RETURN
600 IFD>9THENB$=B$+CHR$(55+D)ELSEB$=B$+CHR$(48+D)
610 RETURN
620 PRINT@448,"ENTER HEX EG. 0000 ";:INPUTH$:H1$=H$:GOSUB220
630 X=USR(0):H$=RIGHT$(H$,4)
640 HH$=LEFT$(H$,2):GOSUB680:E=H*256:E1=H
650 HH$=RIGHT$(H$,2):GOSUB680:E=E+H:E2=H
660 PRINT@42,H1$:D=E:C=E:B$="":PRINT@105,E2:PRINT@121,E1
670 RETURN
680 H=0:A$=LEFT$(HH$,1):GOSUB700:H=D*16
690 A$=RIGHT$(HH$,1):GOSUB700:H=H+D:RETURN
700 D=ASC(A$):IFD<58THEND=D-48ELSED=D-55
710 RETURN
720 B$="":PRINT@448,"ENTER LSB.";:INPUTS:GOSUB220
730 PRINT@448,"ENTER MSB.";:INPUTM:GOSUB220:X=USR(0)
740 D=S+256*M:C=D:S=0:M=0:GOSUB420:GOSUB540:GOSUB260:RETURN
```

VZ MODS INFORMATION BY ROSS WOODS

M O D I F Y I N G V Z 200 / 300 16 K R A M E X P A N S I O N

Use of the VZ200 RAM PACK will give only 6K EXTRA to the VZ 300 COMPUTER. To get the FULL 16K with VZ300 we refer you to an article by STEVE OLNEY in ETI, FEBRUARY 1986, (Page 71-74). The modification involves "PIGGYBACKING" of an extra IC, a (74LS00) and cutting of ONE PCB TRACK. (It is possible to fit a SWITCH SO THAT IT CAN BE USED WITH BOTH VZ 200 / 300 COMPUTERS).

On the other hand you may want to use a VZ300 RAM PACK with a VZ200 COMPUTER, the details of this modification is normally supplied together with the instructions for the RAM PACK from DICK SMITH ELECTRONICS. It involves moving LINKS on the PCB. (It may be possible to fit a SWITCH SO THAT IT COULD BE USED WITH BOTH VZ 200 / 300 COMPUTERS).

I N C R E A S I N G V Z 200 / 300 I N T E R N A L M E M O R Y

SUPER II VZ200 MODIFICATION - ETI PROJECT 687

The SUPER II 34K MEMORY BOARD REPLACES THE VZ 200 INTERNAL 6K MEMORY BOARD. The new Top of MEMORY now is 'FFFF'. This Project was PUBLISHED in ETI JULY 1986, Pages 55-60 by MATTHEW SORREL. It uses 4 x 8K & 1 x 2K MEMORY chips on a PCB REPLACING THE VZ200 INTERNAL 6K MEMORY PCB.

FOR those who have a VZ 300 and require FULL EXPANSION to Top of MEMORY 'FFFF' there was an article in ISSUE Number 3, Page 12-13 by Dave Boyce in the Hunter Valley VZ Users' Group Newsletter (Now known as HUNTER VALLEY VZ JOURNAL).

Write to Dave BOYCE or H.V.VZ Users' group for more information.

ENQUIRIES RE ETI BACK ISSUES OR PHOTOSTATS OF ARTICLES WRITE TO
ETI READER SERVICES, 180 BOURKE RD. ALEXANDER N.S.W. 2015.
OR PO BOX 227 WATERLOO, N.S.W. 2017.

COST OF BACK ISSUES	$4.00	IF AVAILABLE.
COST OF PHOTOSTATS	$4.00	(ARTICLE IN 1 ISSUE).
COST OF PHOTOSTATS	$8.00	(ARTICLE IN 2 OR MORE ISSUES).

MAZE OF DOOM

PART 2

```
450 PRINT"YOU ARE AT A 'T' JUNCTION."
451 PRINT"YOU MAY GO NORTH, SOUTH OR WEST"
452 PRINT"WHICH WAY";:GOSUB5000
453 IFD$="N"THENGOTO830
454 IFD$="S"THENGOTO410
455 IFD$="W"THENGOTO440
456 GOSUB999:GOSUB5000:GOTO453
460 ' GORGON!
461 IFGH=1THEN468ELSEPRINT"A STRANGE LOOKING BEING IS ":GOSUB965
462 PRINT"UNCURLING ITSELF.  SUDDENLY A":GH=1
463 PRINT"WOMAN'S HEAD LOOKS AT YOU AND A"
464 PRINT"LARGE NUMBER OF HISSING SNAKES"
465 PRINT"WRITHES AROUND THE HEAD!  "
466 PRINT"IT'S A GORGON!!!!!"
467 PRINT"YOU MUST FIGHT!: <RETURN>":INPUTT$:GOSUB1000:RETURN
468 PRINT"THE REMAINS OF THE GORGON LIES  WHERE YOU LEFT IT."
469 PRINT"THE ROOM IS OTHERWISE EMPTY.":RETURN
470 PRINT"YOU FOLLOW A LONG PASSAGE WHICH"
471 PRINT"FINALLY ENDS IN A FILTHY ROOM."
472 PRINT"IN THE CORNER LIES A SHAPE":IFGR=1THEN477
473 PRINT"WHICH RISES AS YOU ENTER....!":FORI=1TO500:NEXTI
474 PRINT"TWO LITTLE PIGGY EYES GLARE AT":GR=1
475 PRINT"YOU... IT'S AN ORC!!! (RETURN)":GOSUB51000:INPUTT$
476 GOSUB967:GOSUB1000:GOTO478
477 PRINT"WHICH YOU RECOGNISE AS THE ORC."
478 PRINT"THERE IS NO OTHER WAY OUT OTHER"
479 PRINT"THAN THE WAY YOU CAME IN.       YOU RETURN.":RETURN
500 REM ****CYCLOPS****
501 PRINT"YOU COME OUT INTO A LARGE ROOM"
502 PRINT"OPEN TO THE SKY!!  THERE IS A":IFCYCL=1THENGOTO519
503 PRINT"COUCH ON ONE SIDE.   LYING ON":CYCL=1
504 PRINT"THE COUCH IS A HORRIFYING "
505 PRINT"FIGURE!!  A SINGLE EYE GLARES "
506 PRINT"AT YOU!!!  ITS A CYCLOPS!!!":GOSUB51000
507 PRINT"HIT RETURN TO FIGHT!!":INPUTT$:GOSUB910:GOSUB1000
509 PRINT"YOU ARE GASPING AFTER THE FIGHT"
510 PRINT"THE CYCLOPS WAS A TERROR!"
512 PRINT"YOU FINALLY SHEATH YOUR SWORD"
513 PRINT"AND STAGGER TO YOUR FEET.":GOTO520
519 PRINT"CYCLOPS'CORPSE HERE."
520 PRINT"THERE ARE FOUR PATHS ON,"
521 PRINT"NORTH, SOUTH, EAST & NORTH-WEST"
522 PRINT"WHICH WAY";:GOSUB5000
523 IFD$="N"THENGOTO560
524 IFD$="S"THENGOTO530
525 IFD$="E"THENGOTO300
526 IFD$="NW"THENGOSUB470:GOTO500
527 GOSUB999:GOSUB5000:GOTO523
530 REM *****THE DESERT!!****
531 PRINT"AFTER A LONG WALK, YOU COME "
532 PRINT"OUT TO AN OPENING INTO THE"
533 PRINT"DESERT!   WILL YOU RETRACE"
534 PRINT"YOUR STEPS, OR GO OUT INTO THE"
535 PRINT"DESERT?(R/D)":GOSUB5000
536 IFD$="R"THENGOTO500
537 PRINT"YOU WALK OUT ONTO THE DESERT."
538 PRINT"THE SUN IS QUITE HOT ON YOUR "
539 PRINT"BODY.   AFTER A SHORT WALK,"
540 PRINT"YOU TURN BACK TOWARDS THE MAZE"
```

Happy Computing in the New Year, to all our friends around the country

CONTINUED ON NEXT PAGE

```
541 PRINT"<<RETURN>>":INPUTT$
542 CLS:PRINT"YOU CANNOT FIND THE ENTRANCE!!!"
543 PRINT"YOU ARE DOOMED TO WALK ON UNTIL"
544 PRINT"YOU DIE FROM SUNSTROKE OR"
545 PRINT"THIRST!!!   THE MAZE OF DOOM "
546 PRINT"HAS CLAIMED ANOTHER VICTIM!!"
547 SOUND30,1:SOUND29,1:SOUND30,7:SOUND0,2:SOUND28,2:SOUND26,1
548 SOUND25,1:SOUND23,1:SOUND22,5:SOUND23,7:GOTO1530
560 REM ****BOTTOMLESS PIT***
561 POKE30744,1
562 CLS:PRINT"YOU ENTER A VERY DARK CAVE!"
563 PRINT"WILL YOU CRAWL OR WALK(C/W)";:INPUTD$
564 IFD$="W"THENGOTO566
565 IFD$="C"THEN580ELSEPRINT"NOPE, TRY AGAIN":INPUTD$:GOTO564
566 PRINT"YOU WALK IN, AND SUDDENLY YOUR"
567 PRINT"FOOT MEETS NOTHING!!!!!":FORI=1TO1000:NEXTI
568 CLS:PRINT@450,"YOU FALL!!!!!"
569 FORI=30TO1STEP-4:SOUNDI,1:PRINT:PRINT:NEXTI
570 CLS:PRINT"YOU HAVE BROKEN ALL YOUR BONES!"
571 PRINT"ANOTHER VICTIM HAS FALLEN FOUL"
572 PRINT"OF THE MAZE OF DOOM!......":GOTO1530
580 PRINT"YOU CRAWL IN SLOWLY AND FEEL "
581 PRINT"YOUR WAY ALONG THE WALL."
582 PRINT"THEN YOU SEE IN THE FAINT LIGHT"
583 PRINT"FROM OUTSIDE, A HUGE PIT!! "
584 PRINT"YOU COULD HAVE FALLEN INTO IT!!"
585 PRINT"THERE ARE TWO WAYS ON,":POKE30744,0
586 PRINT"SOUTH AND EAST";:GOSUB5000
587 IFD$="E"THEN590
588 IFD$="S"THEN500ELSEGOSUB999:GOSUB5000:GOTO587
590 PRINT"YOU ARE AT A 'T' JUNCTION."
591 PRINT"YOU MAY GO NORTH,SOUTH OR WEST";:GOSUB5000
592 IFD$="N"THENGOTO750
593 IFD$="S"THENGOTO700
594 IFD$="W"THENGOTO560
595 GOSUB999:GOSUB5000:GOTO592
600 REM ****MINI-MAZE****
601 PRINT"YOU SET OFF AND FIND THE PATH"
602 PRINT"TWISTS AND TURNS, MEETING UP"
603 PRINT"WITH OTHER PATHS.  YOU FIND"
604 PRINT"THAT YOU ARE LOST!!!"
605 PRINT"PATHS LEAD IN ALL DIRECTIONS!!"
606 PRINT"TYPE IN N,S,E,W, UP OR DOWN.";:INPUTT$
607 FORI=1TO10:INPUTT$
608 PRINT"YOU ARE IN THE MAZE";:NEXTI
609 INPUTT$:PRINT"YOU'RE OUT!!!"
610 LETOP=RND(10)
611 IFOP>8THENGOTO530
612 IFOP<6THENGOSUB730:PRINT"YOU GO BACK INTO THE MAZE."
613 IFOP<6THENGOTO607
620 PRINT"YOU FIND YOUR WAY TO SAFETY":GOTO750
630 REM ****DRAGON!!*****
631 PRINT"AFTER A SHORT WALK, YOU TURN A"
632 PRINT"CORNER AND A ROAR MAKES THE ":GOSUB920
633 PRINT"WALLS ROCK!!!!":SOUND1,5
634 PRINT"IT'S A DRAGON!!!!!":GOSUB51000
636 PRINT"YOU ARE FEELING VERY TIRED, "
637 PRINT"BUT YOU MUST COMBAT IT!"
638 PRINT"HIT RETURN!!":INPUTT$
639 DRAG=1
640 GOSUB1000:GOTO654
```

CONTINUED ON NEXT PAGE

```
644 PRINT"YOU POINT YOUR HANDS AT THE"
645 PRINT"DRAGON AND A BOLT OF LIGHTENING"
646 PRINT"STRIKES IT!!!   BUT DRAGONS ARE"
647 PRINT"HARDY BEASTS, AND YOU ONLY "
648 PRINT"WOUND IT.   IT'S SKILL IS BADLY"
649 PRINT"DIMINISHED!  YOU MUST STILL "
650 PRINT"FIGHT IT, BUT YOUR FIGHT WILL"
651 PRINT"BE SO MUCH EASIER!  <<RETURN>>";:INPUTT$
652 Z$="DRAGON":ZZ=5:GOSUB1025
654 PRINT"YOU ARE ALMOST EXHAUSTED WITH"
655 PRINT"THIS LAST BATTLE, BUT SOMETHING"
656 PRINT"TELLS YOU THAT YOU ARE ALMOST "
657 PRINT"AT THE END OF YOUR QUEST!!":DRAG=0
658 PRINT"AHEAD THERE IS ONE DOOR.   "
659 PRINT"HIT RETURN TO ENTER IT!!";:INPUTT$
660 GOTO 760:REM WARLOCK!
670 PRINT"YOU FOLLOW A TWISTING PATH"
671 PRINT"FOR A WHILE, AND THEN"
672 PRINT"SUDDENLY YOU COME OUT INTO A "
673 PRINT"LARGE ROOM LITTERED WITH BONES"
674 PRINT"AND SCRAPS OF HUMAN FLESH!"
675 PRINT"IN THE MIDDLE OF IT ALL SITS"
676 PRINT"THE STRANGEST LOOKING CREATURE"
677 PRINT"YOU HAVE EVER SEEN!!!  WHEN IT"
678 PRINT"SEES YOU, IT LEAPS AT YOU WITH":GOSUB971
679 PRINT"A GROWL!!!   IT'S A GHOUL!!!":GOSUB51000
680 PRINT"YOU MUST FIGHT!! HIT RETURN";:INPUTT$:GOSUB1000
681 PRINT"THERE IS NO OTHER EXIT, OTHER"
682 PRINT"THAN THE DOOR YOU ENTERED BY."
683 PRINT"YOU RETURN.":RETURN
700 REM ****DWARF ATTACK!!***
701 PRINT"YOU COME OUT INTO A CANYON"
702 PRINT"WITH HIGH WALLS ON BOTH SIDES."
703 GOSUB905
704 PRINT"AS YOU WALK ALONG THE BOTTOM,":IFDWARF=1THENGOTO720
705 PRINT"YOU CAN HEAR ODD SOUNDS ABOVE":LET DWARF=1
706 PRINT"YOU IN THE ROCK FACE.....":FORI=1TO1000:NEXTI
707 PRINT"SUDDENLY A DWARF LEAPS DOWN"
708 PRINT"AND ATTACKS!!!   IT'S AN"
709 PRINT"     AMBUSH!!!!":GOSUB51000
710 PRINT"HIT RETURN TO FIGHT THE DWARF!";:INPUTT$:GOSUB1000
711 PRINT"THE DWARF LIES DEAD!"
712 PRINT"YOU MAY CONTINUE.":GOSUB15050
713 PRINT"YOU MAY GO NORTH OR SOUTH.";:GOSUB5000
714 IFD$="N"THENGOTO590
715 IFD$="S"THENGOTO440
716 GOSUB999:GOSUB5000:GOTO714
720 PRINT"YOU REALISE YOU ARE BACK IN THE"
721 PRINT"DWARF CANYON.   YOU PASS THE "
722 PRINT"DWARF'S BODY LYING WHERE YOU"
723 PRINT"LEFT IT.":GOTO713
730 PRINT"YOU ARE IN THE MIDDLE OF THE"
731 PRINT"MAZE, AND TOTALLY LOST, WHEN "
732 PRINT"SUDDENLY YOU COME OUT INTO A "
733 PRINT"LARGE ROOM LITTERED WITH BONES"
734 PRINT"AND SCRAPS OF HUMAN FLESH!"
735 PRINT"IN THE MIDDLE OF IT ALL SITS"
736 PRINT"A HUGE SHAMBLING FIGURE...":FORI=1TO1000:NEXTI
737 PRINT"IT'S THE MONITAUR!!!  WITH A "
738 PRINT"ROAR THE MONSTER LUNGES AT YOU,"
739 PRINT"AND YOU HAVE NO CHOICE!":GOSUB51000
```

PART 3 NEXT ISSUE

IS THIS A VIDEO DISK ??

By Robert Quinn

Those VZ users that have managed to acquire a disk drive and have studied the VZ DOS (Disk operating system) will have discovered how primitive and limited the sequential data file handling facilities provided for the VZ are.
They are as useless as the DOS manual indicates. I avoid them like the plague.
So what do you do when you want to save data to disk?.
Use the binary file commands BSAVE and BLOAD.

The DOS manual is quite misleading, as to the potential uses of these commands.
You are told that " These commands are for binary language programs", which gives the impression that they are to be used exclusively to save and load machine code programs and so only of interest to those users who are into machine language programming. Not so!!.
You don't have to know anything about machine code, assembly language, and all that, to make excellent use of these binary commands.

BSAVE faithfully records to disk the contents of all the memory cells between the START address and the END address, whether the block of memory contains a machine code program, a basic program, or any kind of data.
BSAVE doesn't query what it means:

MINE NOT TO REASON IT,
MINE JUST TO SAVE IT BIT BY BIT.

BSAVE also keeps track of where, in memory, the file came from.

BLOAD simply reverses the BSAVEing process: it faithfully transfers the file, bit by bit, back into you computers memory, right back into the very same cells it originally came from.
Knowing this, and with a bit of imagination, all kinds of uses for these commands emerge. Here is one of the more obvious :-

The screen in MODE(0), low res text mode occupies 512 bytes of memory from 28672 (7000hex) to 29183 (71FFhex).
So, to save on disk the entire contents of the screen, you would :-

```
BSAVE"filename",7000,71FF
```

BLOADing this file will return all the data in the file back to screen memory from whence it came - instant recall of the screen text/pattern from disk.
In a program you would, of course, have BSAVEing and BLOADing subroutines so as to avoid having to BSAVE the BSAVEing command on the screen or having a READY message appear after BLOADING, using INKEY$ to call the subroutines.

Any kind of data can be placed in some reserved memory (by POKEing, for example) and the data BSAVEd from reserved memory to disk.
The minor irritant is that you have to know the START and END addresses in HEX.
So here is a brief decimal to hex conversion program :-

```
1 N$="0123456789ABCDEF"
2 INPUT"DEC#";M
3 C%=M/4096:M=M-4096*C%:GOSUB7
4 C%=M/256:M=M-256*C%:GOSUB7:C%=M/16:M=M-16*C%:GOSUB7
5 C%=M:GOSUB7
6 PRINT"= HEX ";D$:D$="":GOTO2
7 D$=D$+MID$(N$,C%+1,1):RETURN
```

< Continued on next page >

What memory can you conveniently reserve for this purpose.?
Well, yourVZ already reserves 1536 bytes of memory which it never uses, so long as you remain in text mode (MODE(0)). I am referring to video memory.

Your VZ reserves a total of 2048 bytes of memory for video display, of which only the first 512 bytes are used to display the screen in text mode.
The remaining video memory - from 29184 (7200hex) to 30719 (77FFhex)
can be used to store any data you like, and it will remain intact in that memory so long as you don't switch to high res mode (MODE(1)).

BREAK or LOAD a program, RUN (text mode only) a program, and any data you have saved in video memory from 29184 on, remains absolutely secure.

SOFTWARE REVIEW

Reprinted with permission, from Hunter valley VZ Journal.

Program Copier from Celestron Software is a handy utility that allows you to copy your own or other peoples copy protected software for them. This is the program you should be using to make those backup copies we're always raving about.

This program, written by N. Sarafoudis, is really excellent and is another program which is extremely well worth investing in.
The program itself consists of three parts :-

Part one - The title page and loader

Part two - The central machine code program

Part three - The basic driver

After loading the program you are presented with a menu containing six options.

Load
Text Saver
Binary Saver
Program information
Tape to Disk Transfer
Quit

Saving a Text program can be done without Program Copier easily enough by just using the basic CSAVE command. But saving a Binary program is a more involved process without the Program Copier.

To save Binary (machine language) programs using Program Copier you must firstly load the program from tape using the Load routine and then simply save back to a tape using the binary Save routine.
If you choose program information, you will be given the start and end addresses of your saved program, and how much memory, in kilobytes, it takes up.

The tape to disk transfer works quite well and overall this is a good program.

NOTE..
Celestron software is no longer operating and I understand that this program is the same as the copier which appears in the Hackers Delight program.
(see review in the December issue of VZ LINK).....

Peter.

PROGRAMMING HINTS

Here is a list of programming Tips that I have found or discovered over the years. Some of these hints are not only applicable to the VZ, but to many other computers.

1. To squeeze more into the small 2 line limit of BASIC programming lines, you can substitute a question mark (?) for PRINT and change all THENs to commas (,).
When listed, the question marks expand to the actual word 'PRINT'.
This expansion may take up more than the allowed two lines, so care must be taken when editing, because anything exceeding the two line limit will be truncated when entered. Changing the PRINT back to a question mark and deleting the space after the line number will allow you to enter it again without any problems.

2. Many people complain about the slow execution times in BASIC programs.
By organising programs in a logical way, execution time may be decreased quite considerably. For example, by putting commonly used subroutines at the start of the program, rather than at the end, speeds up programs. Similarly, putting routines used least at the end, such as instructions also helps.

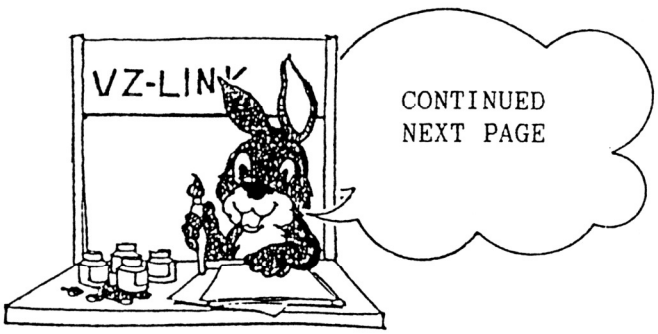
3. Keeping variable names to a two character recognition limit means the computer does not have to read more characters than necessary. This also saves memory and space in the 2 line limit. Since the computer only recognises the first two characters of the variable name, it is wasteful (apart from human use) to use longer variable names.

4. Using fewer variables also speeds up execution. When variables are first assigned, they are placed in a look-up table, along with their values, for future reference. The more variables in the table, the further the computer must look for a variable, and therefore, the longer it takes to find a value. So try to use, where possible, the same variable names in FOR/NEXT loops, INKEY$ statements etc.
The fewer variables defined, the quicker the machine will find a variables value.

5. Defining commonly used variables first can also make a marked difference in program speed. For example, if you are writing a CAR RACE type program, it would be a good idea to define the car position variables first, before any other variables are mentioned. The computer will then place these on top of its look up table, so it will find them first when it goes to find a variables value.

6. Keywords such as IF, THEN, RESTORE, RETURN all take up one byte of memory.
All other characters, including the remarks after REM statements and spaces, also take up one byte of memory. So deleting all unnecessary spaces and all REM statements can save a lot of memory, especially if you are prone to separate each word by a space and document your program with multitudes of REMs.

7. When used in conjunction with THEN, GOTO's may be deleted.
Similarly THEN's may be deleted when followed by a PRINT. This may allow you to squeeze more onto a line.
One thing to watch is ELSE's because each ELSE must have a corresponding THEN.



8. Programmers wanting to store data to tape in programs such as in business or database applications tend to lose interest when they find out how slow this method is. Take the following subroutine which places 60 pieces of data in array D, on to cassette :-

```
100 FOR A = 1 TO 60
110 PRINT#"DATA",D(A)
120 NEXT A
```

When the computer executes this, it will send about three seconds of tape formatting header for the computer to recognise on loading, then send the filename and finally the data. It will repeat this 60 times!. A much better subroutine would be :-

```
100 FOR A = 1 TO 60 STEP 6
110 PRINT#"DATA",D(A),D(A+1),D(A+2),D(A+3),D(A+4),D(A+5)
120 NEXT A
```

Now it will only send out a header 10 times, instead of the 60 times previously, saving about 3 minutes!.

9. If you have ever tried to animate shapes in 'Hi-Resolution' graphics using SET and RESET, then you will realise how slow this method is.
A much better method is to POKE the graphics into screen memory. This method is generally much quicker when drawing big blocks of characters. It also tends to use less memory.
The screen memory in Hi-Resolution MODE(1) occupies addresses from 28672 to 30720.
It is easy to follow if you are familiar with binary. That is :
1 byte is made of 8 bits, each bit being a zero or a one.

one byte is :

7	6	5	4	3	2	1	0

To convert from binary to decimal, multiply each bit by the power of 2

eg : 10000101 = 2^7+2^3+2^1=133 decimal

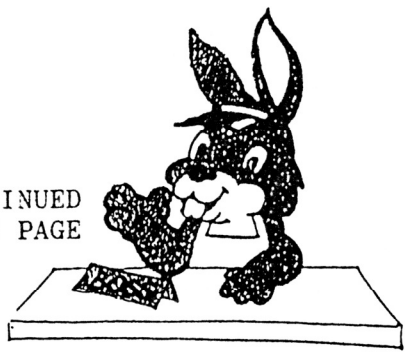
So what has this to do with the graphic screen?. Four pixels are controlled by each byte of screen memory. It is easier to understand by breaking the byte into separate bits.

Bit 7 and Bit 6 make up pixel 0. Bit 5 and Bit 4 make up pixel 1.
Bit 3 and Bit 2 make up pixel 2. Bit 1 and Bit 0 make up pixel 3.

The reason for two bits (1/2 a nibble ?) is the colour selection. It depends on what background colour you are using to select which half of the colours you get.

Green = Background 0 + bit pattern 00
Yellow = Background 0 + bit pattern 01
Blue = Background 0 + bit pattern 10
Red = Background 0 + bit pattern 11
and
Buff = Background 1 + bit pattern 00
Cyan = Background 1 + bit pattern 01
And so on.

CONTINUED
NEXT PAGE



9. Continued.

Each line of pixels takes up 128/4 = 32 bytes of memory.
The screen looks as follows :-

```
Top Left 28672:28673............28702:28703 Top Right
         28704:28705............28734:28735
           :     :        :       :     :
           :     :        :       :     :
           :     :        :       :     :
         30657:30658............30687:30688
Bottom Left 30689:30690............30719:30720 Bottom Right
```

So to draw a line in yellow around the border :-

```
10 MODE(1):COLOR,0
20 FOR A = 28762 to 28703 : REM top line addresses
30 POKE A,85 : REM bits 6+4+2+0=64+16+4+1
40 POKE A+2016,85 : REM bottom line address is 2016 more than the top line
50 NEXT A
60 FOR A = 28704 TO 30657 STEP 32 : REM the left hand side from 1 line below
top to 1 line above bottom
70 POKE A,64 : REM bit 6 only
80 POKE A+31,1 : REM right side is 31 bytes more than left, bit 0 only
90 NEXT A
```

These tips should help to speed up programs and make them more memory efficient.

If you have any other hints you may have discovered, then send them to peter.

REPRINTED from XILOG Newsletter.

USR FUNCTION EXPLAINED

By Bob Kitch.

The USR function is one of the most useful BASIC commands.
(the others being GOSUB/RETURN and DEF FNA (not implemented on the VZ))
But unfortunately the USR function is poorly understood - perhaps because of poor documentation in the manual.
The USR function establishes a LINK between a BASIC program and a machine code routine situated anywhere in memory. It is used to speed up certain procedures that are slow in interpreted BASIC (largely overcome in compiled BASIC) or to call ROM routines eg.. BEEP/SOUND etc.

The machine language routine is usually entered into memory from DATA statements, but may be previously loaded from a Editor Assembler or some other method, including called from ROM. Furthermore it is best to reserve an area at the top of memory (TOM) to load in the routine.
BASIC string space and the stack then sit below this " reserved " area. These are reset by the CLEAR statement. After the TOM is lowered. This is by far and wide the safest way to load M/L routines with BASIC.
(Others methods are given in the TECHNICAL Manual)

When BASIC encounters the USR statement, it needs to have a pointer to indicate where the M/L routine commences. (Similarly the M/L routine needs to end in a RET statement so that an orderly resumption of BASIC takes place).
The pointers for the start are set into 30862 and 30863 in typical LSB and MSB byte of address.

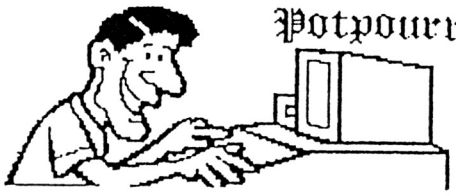
It is also possible to pass a variable from BASIC to the routine. The value enclosed in the ( ) is passed to locations 31009 and 31010. This feature is not often used - but is very powerful.

< CONTINUED on NEXT PAGE >

So much for the outline. The best way to understand all this is to follow an example.

```
10 '*************************
20 '***  2000 VZ SCREENS  ***
30 '***    VERSION 1.2    ***
40 '***   R.B.K. 18/5/86  ***
50 '*************************
60 '
100 '***FIND TOP OF MEMORY.
110 M1=PEEK(30898):L1=PEEK(30897):'***PRESERVE TOM POINTERS.
120 TM=M1*256+L1-20                 :'***RESERVE TOP 20 BYTES.
130 MS=INT(TM/256):LS=TM-MS*256
140 POKE 30898,MS:POKE 30897,LS
150 '
200 '***SET UP LOADING OF USR() ROUTINE.
210 TM=TM+1                         :'***NEXT ADDR IN RESERVED MEM.
220 MS=INT(TM/256):LS=TM-MS*256
230 POKE 30863,MS:POKE 30862,LS
240 AD=TM+10                        :'***ADDR. FOR CHARACTER BYTE.
250 IF TM>32767 THEN TM=TM-65536 :'***CONVERT TO SIGNED INTEGER.
260 IF AD>32767 THEN AD=AD-65536
270 '
300 '***LOAD MACHINE CODE.
310 FOR ID=TM TO TM+13
320   READ VL:POKE ID,VL
330 NEXT
340 '
400 '***Z-80 BLOCK MOVE SUBROUTINE.
410 DATA 33,0,112       :'LD HL,7000H  (#28672D START VIDEO RAM)
420 DATA 17,1,112       :'LD DE,7001H  (#28673D NEXT OR DEST.)
430 DATA 1,255,7        :'LD BC,07FFH  (#2047D SIZE OF VIDEO RAM)
440 DATA 54,85          :'LD (HL),55H  (#85D YELLOW OR CHAR."U")
450 DATA 237,176        :'LDIR         (BLOCK MOVE INSTRUCTION)
460 DATA 201            :'RET
470 '
500 '***INITIALIZE DELAYS - CONTROL SPEED OF EXECUTION BY D.
510 T=0        :'***TONE  0 IS REST.       RANGE IS 0 TO 31
520 D=4        :'***DURATION  9 IS LONG.   RANGE IS 1 TO 9
530 P=30744    :'***ADDR. FOR INVERSE CONTROL.
540 POKE P,0   :'***SET UP SCREEN.
550 '
600 '***SET UP DEMO LOOP.
610 FOR ID=0 TO 255
620   POKE AD,ID                    :'***SET CHARACTER BYTE.
630   '***SCREEN MESSAGE.
640   MODE(0)                       :'***SET *A/G LO.
650   POKE P,0                      :'***SET INV LO.
660   PRINT@234," CHAR = ";ID:SOUND T,D
670   '***LO-RES SCREENS.
680   '***LO-RES GREEN CHARACTER ON BLACK BACKGROUND.
690   X=USR(0):COLOR,0:SOUND T,D:'***SET CSS LO.
700   '***LO-RES ORANGE CHARACTER ON BLACK BACKGROUND.
710   COLOR,1:SOUND T,D             :'***SET CSS HI.
720   POKE P,1                      :'***SET INV HI.
730   '***LO-RES BLACK CHARACTER ON GREEN BACKGROUND.
740   X=USR(0):COLOR,0:SOUND T,D:'***SET CSS LO.
750   '***LO-RES BLACK CHARACTER ON ORANGE BACKGROUND.
760   COLOR,1:SOUND T,D             :'***SET CSS HI.
770   '***HI-RES SCREENS.
780   MODE(1)                       :'***SET *A/G HI.
790   POKE P,0                      :'***SET INV LO.
800   '***HI-RES COLOR SET 0 - GREEN SURROUND.
810   X=USR(0):COLOR,0:SOUND T,D:'***SET CSS LO.
820   '***HI-RES COLOR SET 1 - BUFF SURROUND.
830   COLOR,1:SOUND T,D             :'***SET CSS HI.
840   POKE P,1                      :'***SET INV HI.
850   '***HI-RES COLOR SET 0.
860   X=USR(0):COLOR,0:SOUND T,D:'***SET CSS LO.
870   '***HI-RES COLOR SET 1.
880   COLOR,1:SOUND T,D             :'***SET CSS HI.
890   '***RESET CONTROLS.
900   POKE P,0:COLOR,0:CLS
910 NEXT
920 '
930 '***RESET TOM POINTERS.
940 POKE 30898,M1:POKE 30897,L1
950 STOP:END
```

Notice that TOM is lowered by 20 bytes.
The M/L set out in DATA statements is POKE'd in.
The DATA statements are set out in " pseudo - Assembler " to make the routine very clear and easy to alter.
30862 and 30863 are set to start of M/L for the USR jump.
Every time X=USR(0) is encountered the M/L routine is executed - the screen fill routine.



from Ken Clarke.

A potpourri of ideas, software modifications, help-line, your ideas, suggestions, gripes, where to find it, cheats, pokes and infinite lives, adventure gaming help-line, hardware help in construction, or in fact anything that other VZ'ers could be interested in or be able to find help for you.....

Computer does a RESET when you least expect it or SCREWS-UP the screen display for no accountable reason. You switch off and then on again and things come back to normal. When this fault crops up, it is usually caused by the contacts at the back of the computer to the add-ons becoming DIRTY and CORRODED. This sets up an INTERMITTENT CONNECTION which should be considered as potentially dangerous to leave unattended. Damage to the internal works could result if left unattended. CLEANING with ISOPROPYL ALCOHOL, METHYLATED SPIRITS or CONTACT CLEANER [but NEVER any other solvent], can bring about the necessary improvement. But a FILM can be left behind. This could be from dissolved varnish, wax etc. CORROSION is best POLISHED or BUFFED clean. What I recommend as the BEST METHOD is a "THREE-FOLD PROCESS".

(i) CLEAN with alcohol or contact cleaner. A "cotton bud" is good for the job.

(ii) BUFF or burnish with an extra soft PENCIL RUBBER.

(iii) A light smear of vaseline. The procedure involves carefully opening the VZ case, preferrably, but can be done via the slots at the back, although rather awkward. It is the computer contacts that give the most trouble - not the accessory add-ons which have better quality, plated, spring contacts. Note that step (i) may be omitted, but the polishing with the rubber and the protective, lubricating film are the two important steps that really do the job properly.

**If you are not sure of doing this job yourself. or would like to be shown how to do it - then contact Peter so arrangements can be made.....

*** VZ-200/300 MAILING LIST ***

- by Ken Clarke.

"DISKLIST" - the Conversion to Disk has a Fatal Flaw!

Merging of Data Files does not work!!! This Bug, has turned into a "screaming foul-up!" that has been bugging a number of members.

I can now - finally - proclaim "EUREKA!" I have found it!

Line 6140 does not allow the program to get back to the Main Menu via Line 868. Instead it shunts the program direct to the Dos Menu.

The "fix-it" is to substitute <RETURN> in place of <GOTO 600> (the rest of the line is OK).

ADDENDUM: Mailing List Enhancement, by Pete Patterson (Issue ≠45).
A typographical error has crept into the listing.
Line ≠755 should be changed to Line ≠775 (the rest of the line is OK.

ADDENDUM GRAPHIC NOTES:

ST GEORGE & the DRAGON. [Issue #40 Feb. 1990] Lines 100 should read 32 shifted <J> ; 121 & 135 shifted <U> ; 180 shifted <I> ; 160, 165, 470, 650, 690 - shifted <G> ; 820 shifted <N>.

WANKEL ROTARY COMBUSTION ENGINE. [Issue #41 March, 1990]
and
DONKEY KONG. [Issue #45 July, 1990]
For both there is a complete re-print available on request, just for the asking. This re-print now displays the graphics characters the correct way and not in negative form.

FOR SALE OR SWAP

BLACK & WHITE TELEVISION / MONITOR

IDEAL FOR VZ 200
BEAUTIFUL PICTURE
WILL SELL OR SWAP FOR COMPLETE DISC DRIVE
Contact Rob Lyons 09 - 5341858